

# Trabalho 1 – Busy Police

22705 - Inteligência Artificial  
2017/2 - Turma C  
Prof. Dr. Murilo Naldi

[naldi@dc.ufscar.br](mailto:naldi@dc.ufscar.br)

# Motivação

- Homenagem ao Atari Busy Police Game



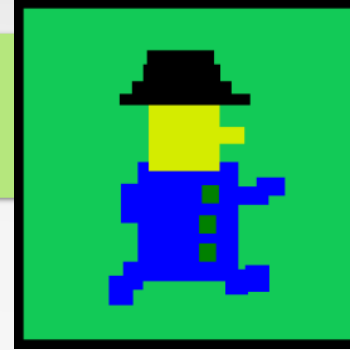
# Motivação

- Policial deve percorrer um supermercado atrás de um ladrão
- O projeto é composto de uma versão estática e simplificada do jogo
- Problema de busca que monta caminho entre policial e ladrão

- Modelagem do ambiente: 5 X 10

[illegible]

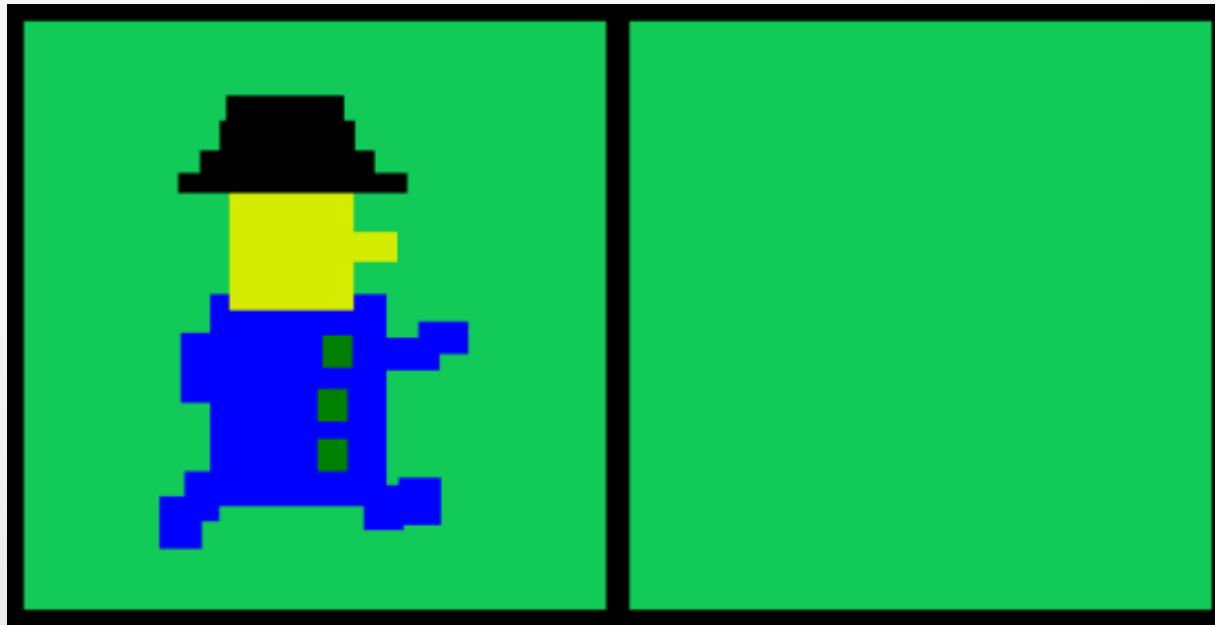
# Agente



- Policial
  - Define a posição inicial do problema de busca
  - Qualquer posição do ambiente
  - Preferencialmente no primeiro andar (1)
- Ladrão
  - Define a posição final do problema de busca
  - Qualquer posição do ambiente
  - Preferencialmente no último andar (5)

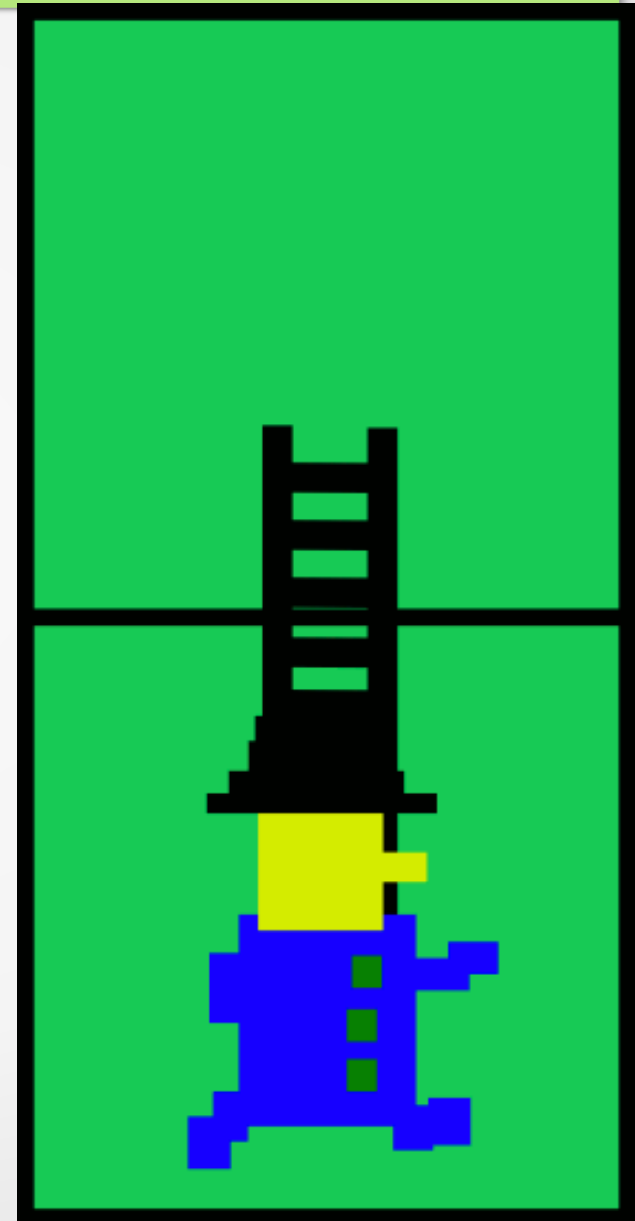
# Movimentação

- Movimentação é livre em um mesmo andar (horizontal)



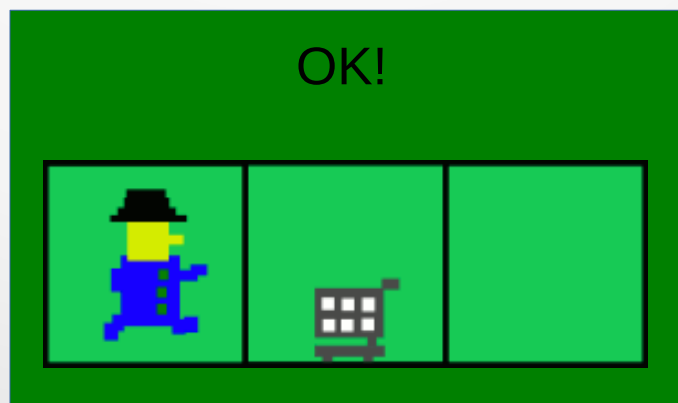
# Movimentação

- Movimentação entre andares só é possível se houver uma escada (vertical)



# Movimentação

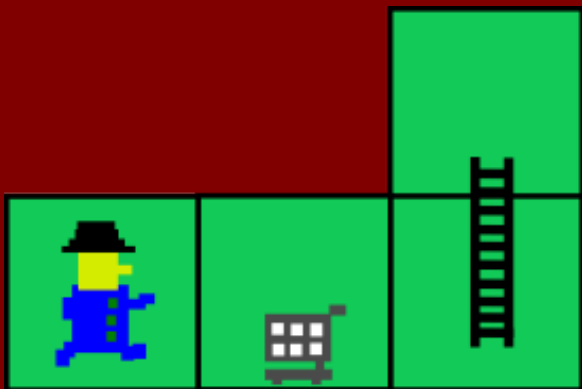
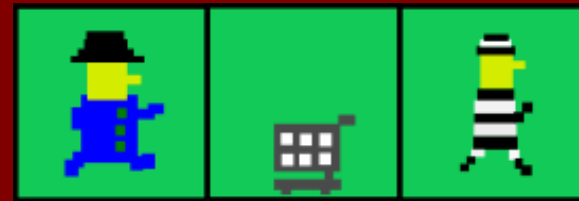
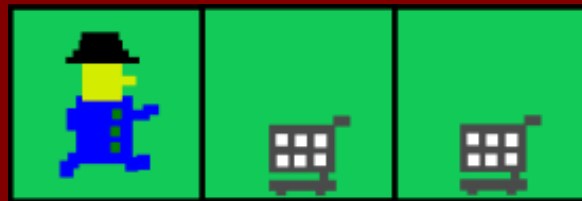
- No jogo, o carrinho faz com que o policial tenha de pular
- No trabalho, o policial só poderá passar por um carrinho se os dois quadrados adjacentes ao carrinho não possuírem nenhum objeto dentro



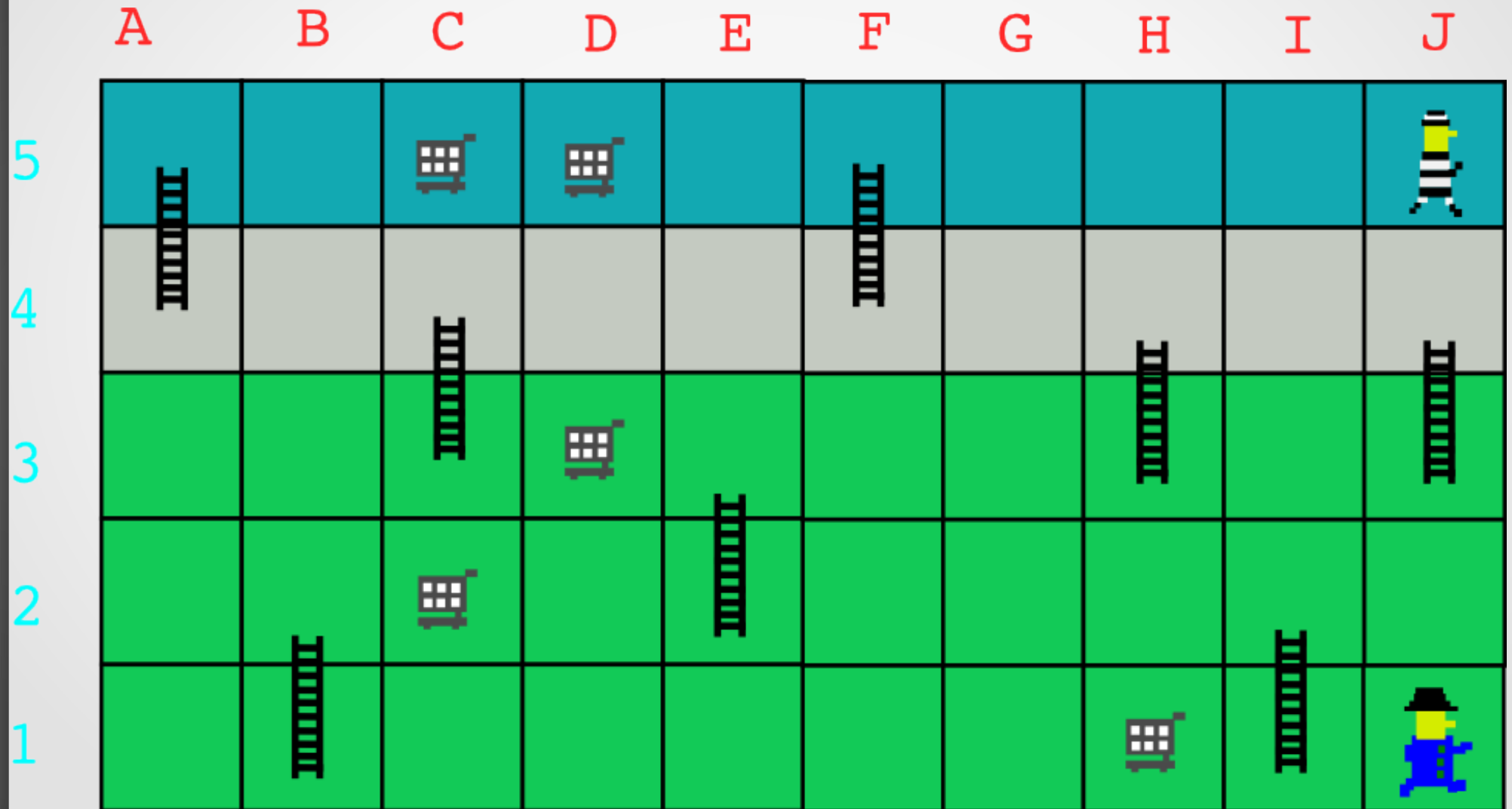


# Movimentação

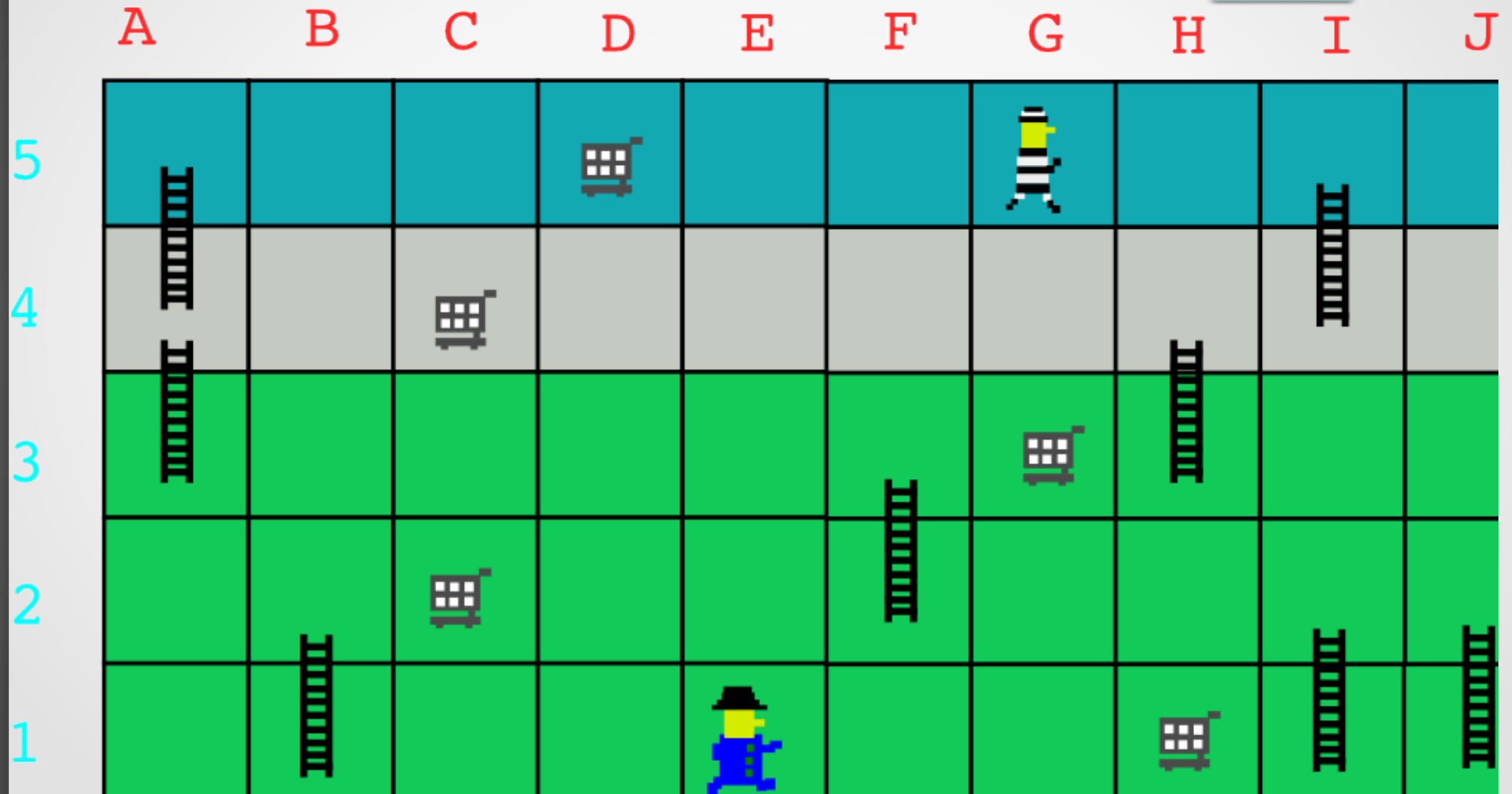
- Alguns exemplos a seguir não permitem a passagem do policial



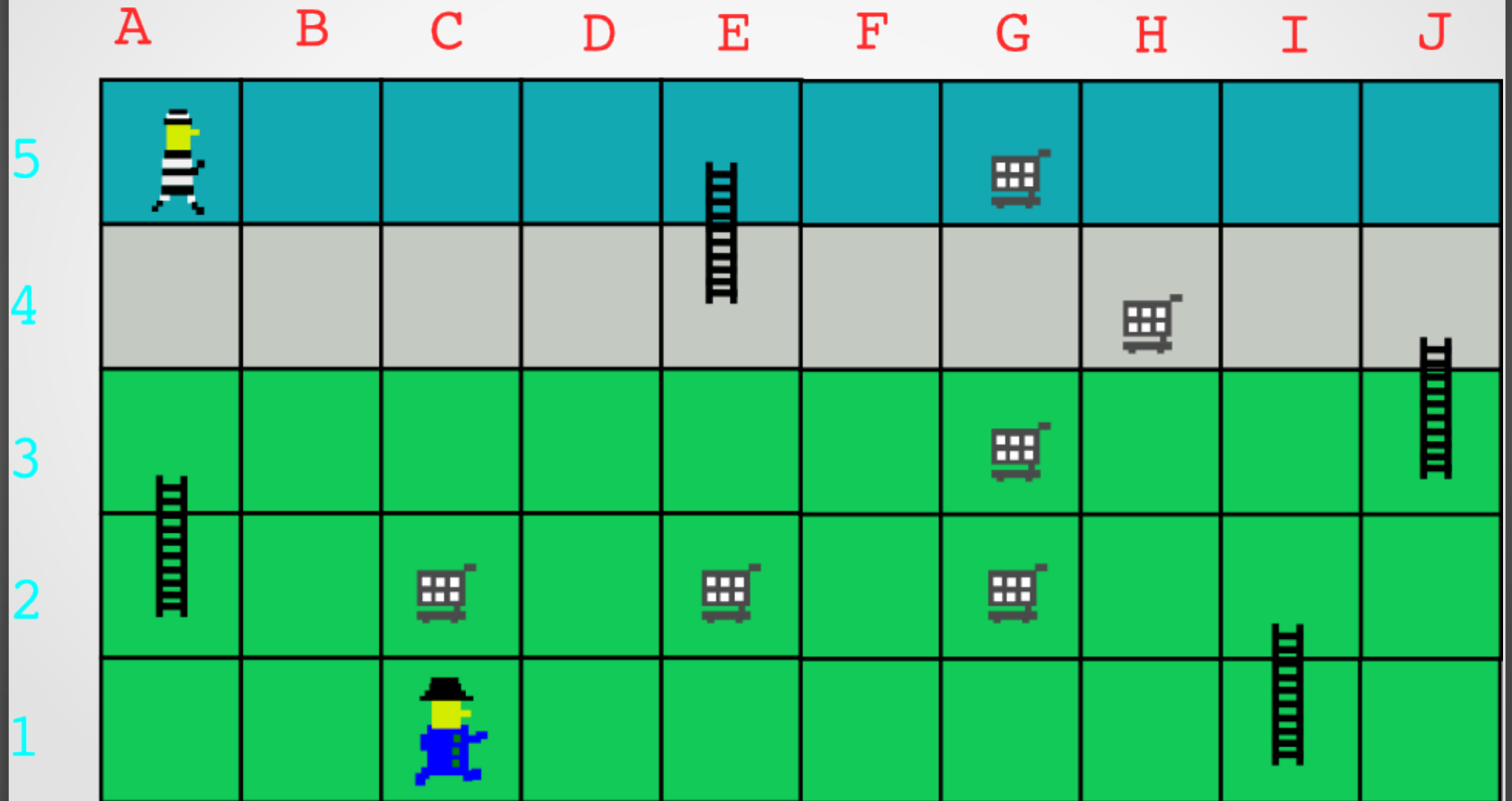
# Ambiente Completo 1



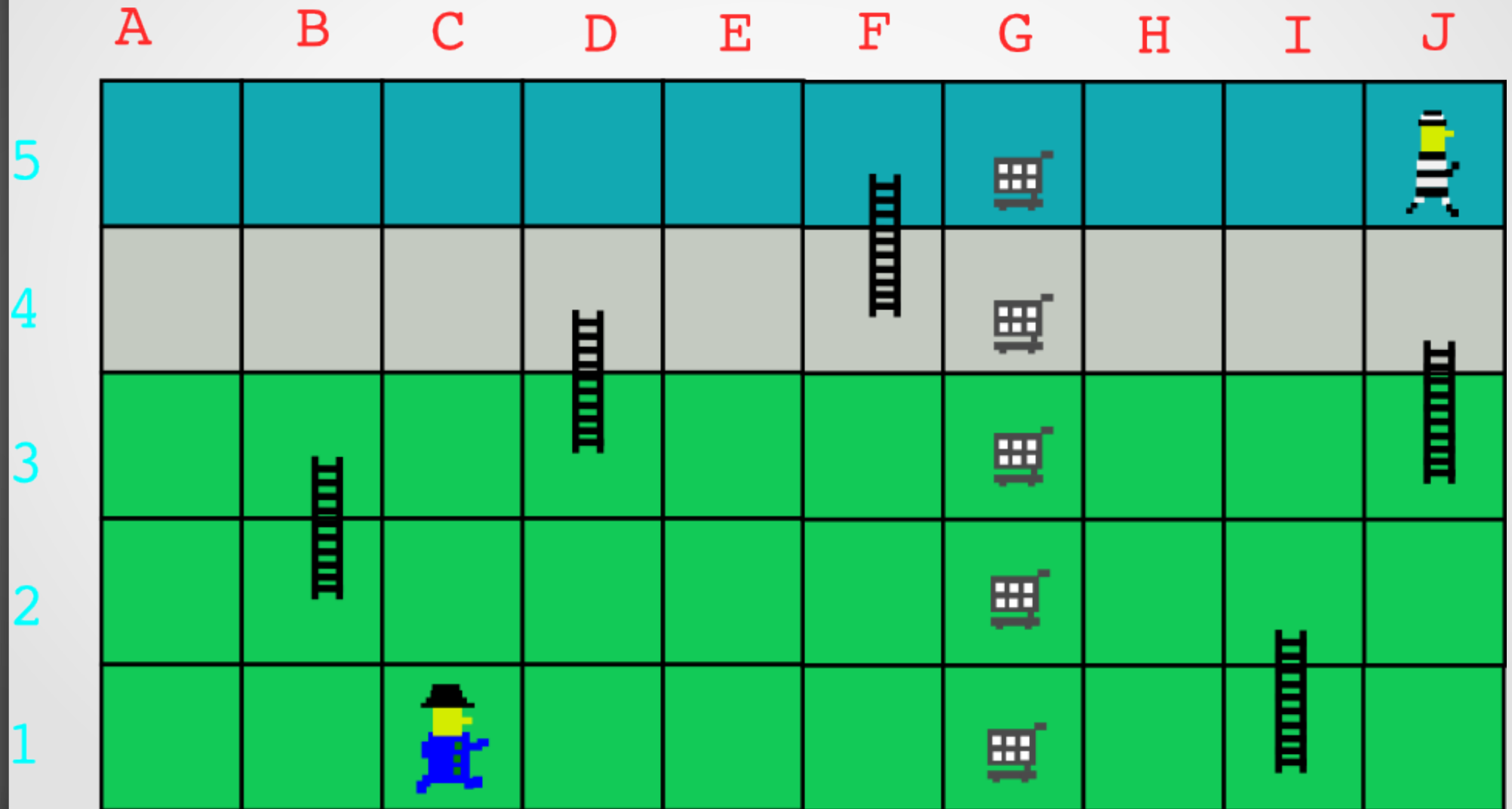
# Ambiente Completo 2



# Ambiente Completo 3



# Ambiente Completo 4



# Objetivo 1

- Implementar em Prolog o ambiente de forma adequada (em regras e fatos), considerando os objetos:
  - Policial e ladrão
  - Escadas
  - Carrinhos de compra
- Deve se considerar que qualquer objeto pode estar em qualquer posição da tabela (programável)

## Objetivo 2

- Dado um estado inicial (policia) e um estado final (ladrao), obter e imprimir um caminho **acíclico** entre os dois, de forma que:
  - O estado inicial seja escolhido pelo usuário
  - Ser capaz de rodar em qualquer cenário, incluindo cenários em que não há caminho até o ladrao

## Objetivo 3 - Bônus Game

- Desenvolver uma funcionalidades extra relativa ao jogo que seja interessante, como por exemplo:
  - Elevador
  - Bolinha
  - Algo que o policial tenha que pegar antes
- Varia com a criatividade e desempenho técnico
- O resultado deve ser desafiador
- Preferencialmente único, ou seja, a marca do grupo!



# Apresentação

- Grupos de até 3 pessoas
  - Identificadas no trabalho (risco de zerrar!)
- Submissão no AVA até 26 de outubro às 14h
  - Apresentação + Artigo Formato SBC + código, os dois primeiros em pdf
- Nota Artigo NR = 0-10
- Nota Apresentação NA = 0-1
- Nota Trabalho NT = NR\*NA

# Apresentação do projeto

- Cada grupo deverá apresentar o seu projeto de maneira didática em 10-15 minutos no dia 26 de outubro.
- A apresentação deve explicar como os fatos e as regras foram gerados, bem como a lógica que faz com que o Prolog consiga chegar no objetivo
- Deve conter exemplo(s) de uso
- Todos os membros do grupo devem apresentar e ganharão notas individuais nessa etapa
- O tempo de apresentação deverá ser dividido igualmente entre os membros
- Serão feitas perguntas durante a apresentação.

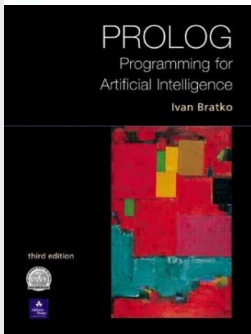
## Não faça!

- Se quiser perder nota:
  - Demore para introduzir o problema em sua apresentação
  - Copie material sem permissão
  - Faça regras de sucessor para cada dupla de estados
  - Não use regras gerais
  - Não mostre que sua solução é capaz de resolver cenários interessantes

# Bibliografia



NICOLETTI, M. C. A Cartilha Prolog. EDUFSCAR. 2005. ISBN 8576000113



Bratko – Prolog: Programming for Artificial Intelligence 2001

Material de apoio no PVAnet

- Prolog para download:  
<http://www.swi-prolog.org/> (ou repositório Ubuntu)